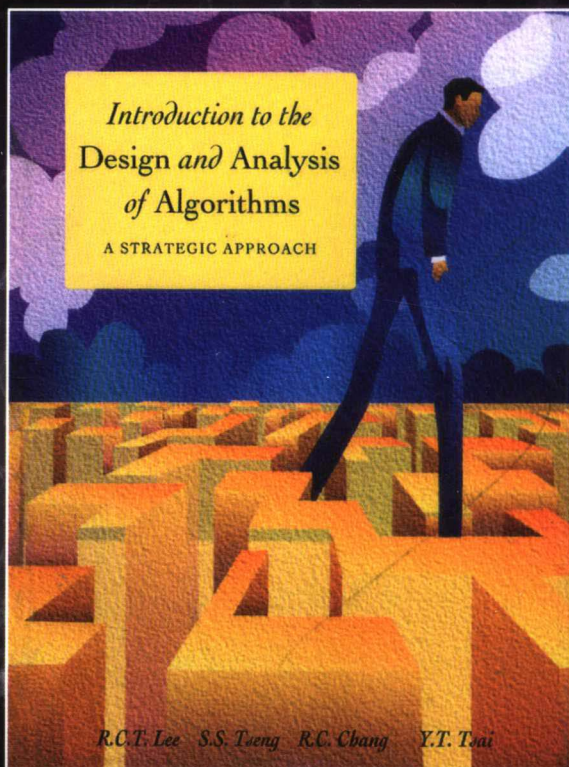




计 算 机 科 学 丛 书

# 算法设计与分析导论

R. C. T. Lee S. S. Tseng R. C. Chang Y. T. Tsai 著 王卫东 译



*Introduction to the Design and Analysis of Algorithms  
A Strategic Approach*



机械工业出版社  
China Machine Press

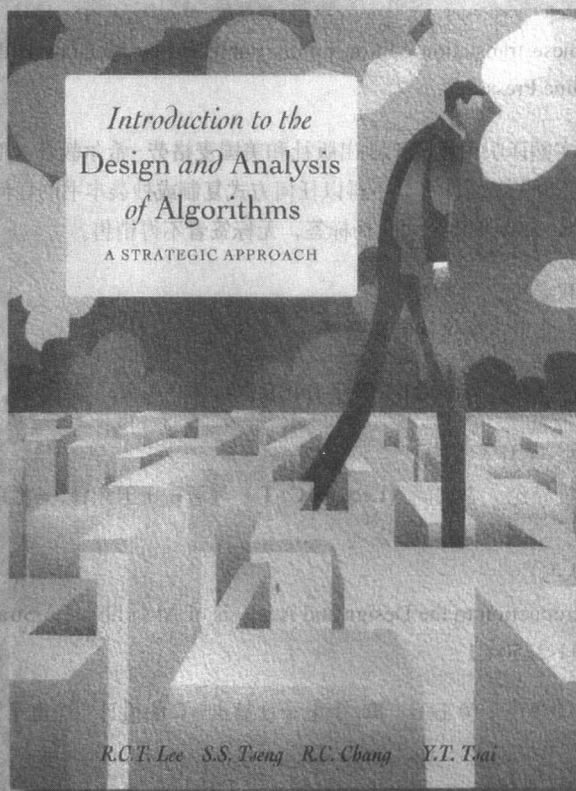
计 算 机 科 学 丛 书

TP301.6/81


2008

# 算法设计与分析导论

R. C. T. Lee S. S. Tseng R. C. Chang Y. T. Tsai 著 王卫东 译



**Introduction to the Design and Analysis of Algorithms**  
A Strategic Approach

 机械工业出版社  
China Machine Press

本书在介绍算法时,重点介绍用于设计算法的策略,非常与众不同。书中介绍了剪枝搜索、分摊分析、随机算法、在线算法以及多项式近似方案等相对较新的思想和众多基于分摊分析新开发的算法,每个算法都与实例一起加以介绍,而且每个例子都利用图进行详细解释。此外,本书还提供了超过400幅图来帮助初学者理解。

本书适合作为高等院校算法设计与分析课程的高年级本科生和低年级研究生的教材,也可供相关科技人员和专业人士参考使用。

R. C. T. Lee, S. S. Tseng, R. C. Chang, Y. T. Tsai; Introduction to the Design and Analysis of Algorithms: A Strategic Approach (ISBN 10: 0-07-124346-1).

Copyright © 2005 by The McGraw-Hill Education (Asia).

Original English edition published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签,无标签者不得销售。

**版权所有,侵权必究。**

**本书法律顾问 北京市展达律师事务所**

**本书版权登记号:图字:01-2007-0469**

**图书在版编目(CIP)数据**

算法设计与分析导论 / 李家同 (Lee, R. C. T.) 等著; 王卫东译. —北京:机械工业出版社, 2007.10

(计算机科学丛书)

书名原文: Introduction to the Design and Analysis of Algorithms: A Strategic Approach  
ISBN 978-7-111-22504-1

I. 算… II. ①李… ②王… III. ①电子计算机-算法设计 ②电子计算机-算法分析  
IV. TP301.6

中国版本图书馆CIP数据核字 (2007) 第154316号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:王 玉

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2008年1月第1版第1次印刷

184mm × 260mm · 24.25印张

定价:49.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

本社购书热线:(010) 68326294

# 出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅开辟了研究的范畴，还揭开了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近260个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”。为了保证这两套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这两套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U.等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、

编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件：[hzjsj@hzbook.com](mailto:hzjsj@hzbook.com)

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

# 专家指导委员会

(按姓氏笔画顺序)

尤晋元  
石教英  
张立昂  
邵维忠  
周克定  
郑国梁  
高传善  
裘宗燕

王 珊  
吕 建  
李伟琴  
陆丽娜  
周傲英  
施伯乐  
梅 宏  
戴 葵

冯博琴  
孙玉芳  
李师贤  
陆鑫达  
孟小峰  
钟玉琢  
程 旭

史忠植  
吴世忠  
李建中  
陈向群  
岳丽华  
唐世渭  
程时端

史美林  
吴时霖  
杨冬青  
周伯生  
范 明  
袁崇义  
谢希仁



# 译者序

本书是算法设计与分析方面的优秀著作。它从策略的角度详细地阐述了关于算法设计的多种设计策略。设计算法的基本原则，也就是策略，显然比算法本身更重要。本书利用算法设计策略作为指导进行算法设计，通过大量实例介绍了设计策略的基本原理，同时强调了对每种算法详细的复杂性分析。

本书的内容具有相当的广度和深度，这包括对各种策略的基本机制、基础算法、基本算法设计技术、算法的复杂度分析等，具体包括算法复杂度与问题的下界、贪心法、分治策略、树搜索策略、剪枝搜索、动态规划、NP完全性理论、近似算法、分摊分析、随机算法和在线算法等。

本书的组织方式简明扼要、循序渐进，涵盖了大多数算法设计中的一般技术，同时包含了剪枝搜索、分摊分析、在线算法以及多项式近似方案等相对较新的思想，这些都是近20年来算法研究中发展迅猛的领域。另外，对于两个重要概念：PTAS和NPO完全性做了清晰的解释。对于在线算法、近似算法、随机算法等结合实例做了详尽的介绍。对于较难理解的分摊分析给出了理论证明。

算法设计一直是备受关注的研究领域之一。因此，本书还介绍了算法设计与分析领域的最新进展、最新成果和特别声明，使读者能够容易地找到进一步研究的方向。

我们相信本书中译本的出版将对国内高校计算机专业算法课程的教学起到积极的推动作用。由于译者水平有限，翻译中难免有错误和不妥之处，恳请读者批评指正。

王卫东

西安电子科技大学计算机学院

2007年7月

# 前 言

研究算法可能有许多原因，其中的主要原因是能够使我们更高效地使用计算机。实践证明，一位不具备很好算法知识的新手，对于其老板及工作单位来说都是一个大麻烦。假如某人要找一棵最小生成树，设想现在或者将来都不存在计算机，那么他通过检查所有可能的生成树来确定最小生成树的做法，对他来说也许足够好了。但是，如果他了解Prim算法，那么有一台IBM个人计算机就足以解决问题。另一个例子是某人要解决语音识别问题，对他来说如何开始是相当困难的。但是，如果他了解能够使用动态规划方法解决的最长公共子序列问题，那么他将会明白该问题是相当容易解决的。算法的研究不仅对于计算机科学家非常重要，而且通信工程师在编码时也要用到动态规划或A\*算法。有许多其他行业的科学家从研究算法中获益匪浅，其中相当一部分是从事分子生物学的科学家。当他们要比较两个DNA序列、两个蛋白质或者三维结构的RNA时，需要了解非常复杂的算法。

此外，研究算法也有很多乐趣。由于我们研究算法已经很长时间，因此在任何时候看到新的设计良好的算法时，或者想出关于设计和分析算法的一些新颖想法时，都会异常兴奋。因此，我们也感到有责任让更多的人共同分享这种乐趣和兴奋。许多看似困难的问题实际上可以通过多项式时间算法解决，而一些看似普通的问题却被证实是NP完全的。最小生成树问题对许多初学者来说似乎是相当困难的，但它却有多项式时间的解决算法。如果稍微对它改动一点就成为旅行商问题，那么它就成为一个NP难问题。另一个例子是3可满足性问题，它是一个NP完全问题。通过降低维数，2可满足性问题将会成为P问题。发现这些事实总是令人着迷的。

在本书中，我们将采取一种与众不同的方法来介绍算法。实际上，我们并不是在介绍算法，而是在介绍用于设计算法的策略。原因很简单：明白用于设计算法的基本原则（也就是策略）显然比算法本身更重要。尽管并非每个算法都基于本书所介绍的某个策略，但大部分是这样的。

剪枝搜索、分摊分析、在线算法以及多项式近似方案都是相对较新的思想，而且是十分重要的思想。众多新开发的算法都是基于分摊分析的，正如我们在关于该主题的章末将看到的那样。

我们将从用来设计多项式时间算法的策略开始介绍。在必须解决看似困难但目前的确还没有多项式时间算法的问题时，我们会介绍NP完全性的概念。尽管经常很难把握它的物理意义，但应用NP完全性的概念还是比较容易的。实际上，其核心思想是为什么每个NP问题实例都关联一个布尔公式集，并且当且仅当该公式集可满足时，该问题实例的回答是“yes”。一旦读者明白了这个思想，就会懂得NP完全性的重要性。我们相信用于解释这个思想而提供的实例将会帮助大多数学生很容易地理解NP完全性的概念。

本书可用作高年级本科生和低年级研究生的教材。依据我们的经验，如果在一个学期（约50学时）内讲授本书，那么无法涵盖书中全部的内容。因此，如果只有一个学期，那么我们建议所有章节都要讲，但不必都完全讲透。不要忽略任何章！关于NP完全性的章是非常重要的，应该让学生理解清楚这章的内容。最难理解的章是第10章（分摊分析），其中涉及许多数学知识。教师在授课时应当非常注意讲解分摊分析的基本思想，而不应过多地讲述相关的数学证明。换句话说，学生应该能理解为什么与好算法相结合的特定数据结构在分摊的意义上执行得非常好。另一个相对困难的章可能是第12章（在线算法）。



许多算法并不容易理解，我们将尽量清晰地介绍这些算法。介绍每个算法时都辅以实例，而每个实例都利用图进行解释。在本书中，提供了超过400幅图来帮助初学者理解。

我们还引用了关于算法设计与分析的大量书籍和论文；并特别声明最新的成果，使读者能够容易地找到进一步研究的方向。“参考文献”中列出了825本书籍和论文，共涉及1 095位作者。

我们还会在适当的时候提供实验的结果。但作为教师，还是应当鼓励学生通过实现算法来验证结论。在每一章的结尾有进一步阅读的论文列表，鼓励学生阅读这些论文以提升他们对算法的理解是非常重要的。希望学生能领会到作者不遗余力地解释大量难以理解的论文是多么辛苦！书中还提供一些用Java编写的程序，便于学生实践。

我们在此不可能列举出在准备本书的过程中帮助过我们的所有人的名字，因为实在是太多了。但他们都属于一个班，是我们的学生或者同事（许多学生后来成为同事）。在每周五晚上的讨论会上，我们总是积极活跃地讨论。这些讨论指出了算法研究中的新方向，也有助于我们决定本书应该包含哪些资料。研究生们一直在追踪大约20种学术期刊，并确保关于算法的每篇重要论文都存储在数据库中，且都关联上关键字。这个数据库对于写作本书是非常有价值的。最后，他们还阅读了本书的手稿，提出了批评意见，并完成了实验。如果没有来自同事和学生们的帮助，我们是不可能完成本书的。在这里，我们对他们所有人表示无限的感谢。

# 目 录

出版者的话  
专家指导委员会  
译者序  
前言

第1章 绪论	1
第2章 算法复杂度与问题的下界	10
2.1 算法的时间复杂度	10
2.2 最好、平均和最坏情况的算法分析	12
2.3 问题的下界	24
2.4 排序的最坏情况下界	25
2.5 堆排序：在最坏情况下最优的排序算法	28
2.6 排序的平均情况下界	33
2.7 通过神谕改进下界	35
2.8 通过问题转换求下界	36
2.9 注释与参考	37
2.10 进一步的阅读资料	38
习题	38
第3章 贪心法	40
3.1 生成最小生成树的Kruskal算法	42
3.2 生成最小生成树的Prim算法	44
3.3 单源最短路径问题	46
3.4 二路归并问题	50
3.5 用贪心法解决最小圈基问题	54
3.6 用贪心法解决2终端一对多问题	56
3.7 用贪心法解决1螺旋多边形最小合作警卫问题	59
3.8 实验结果	61
3.9 注释与参考	62
3.10 进一步的阅读资料	62
习题	63
第4章 分治策略	65
4.1 求2维极大点问题	66
4.2 最近点对问题	67
4.3 凸包问题	69

4.4 用分治策略构造Voronoi图	71
4.5 Voronoi图的应用	77
4.6 快速傅里叶变换	79
4.7 实验结果	82
4.8 注释与参考	82
4.9 进一步的阅读资料	83
习题	83
第5章 树搜索策略	85
5.1 广度优先搜索	87
5.2 深度优先搜索	88
5.3 爬山法	89
5.4 最佳优先搜索策略	90
5.5 分支限界策略	91
5.6 用分支限界策略解决人员分配问题	93
5.7 用分支限界策略解决旅行商优化问题	95
5.8 用分支限界策略解决0/1背包问题	99
5.9 用分支限界方法解决作业调度问题	102
5.10 A*算法	106
5.11 用特殊的A*算法解决通道路线问题	110
5.12 用A*算法解决线性分块编码译码问题	113
5.13 实验结果	115
5.14 注释与参考	116
5.15 进一步的阅读资料	116
习题	117
第6章 剪枝搜索方法	119
6.1 方法概述	119
6.2 选择问题	119
6.3 两变量线性规划	121
6.4 1圆心问题	128
6.5 实验结果	132
6.6 注释与参考	133
6.7 进一步的阅读资料	133
习题	133
第7章 动态规划方法	134
7.1 资源配置问题	137

7.2 最长公共子序列问题	138	9.14 进一步的阅读资料	251
7.3 2序列比对问题	140	习题	252
7.4 RNA最大碱基匹配问题	143	第10章 分摊分析	254
7.5 0/1背包问题	150	10.1 使用势能函数的例子	254
7.6 最优二叉树问题	151	10.2 斜堆的分摊分析	256
7.7 树的带权完全支配问题	154	10.3 AVL树的分摊分析	259
7.8 树的带权单步图边的搜索问题	159	10.4 自组织顺序检索启发式方法的分摊分析	261
7.9 用动态规划方法解决1螺旋多边形 m守卫路由问题	163	10.5 配对堆及其分摊分析	264
7.10 实验结果	165	10.6 不相交集合并算法的分摊分析	275
7.11 注释与参考	165	10.7 一些磁盘调度算法的分摊分析	284
7.12 进一步的阅读资料	166	10.8 实验结果	289
习题	167	10.9 注释与参考	290
第8章 NP完全性理论	169	10.10 进一步的阅读资料	290
8.1 关于NP完全性理论的非形式化讨论	169	习题	290
8.2 判定问题	170	第11章 随机算法	291
8.3 可满足性问题	171	11.1 解决最近点对问题的随机算法	291
8.4 NP问题	176	11.2 随机最近点对问题的平均性能	293
8.5 库克定理	177	11.3 素数测试的随机算法	295
8.6 NP完全问题	184	11.4 模式匹配的随机算法	297
8.7 证明NP完全性的例子	186	11.5 交互证明的随机算法	300
8.8 2可满足性问题	201	11.6 最小生成树的随机线性时间算法	302
8.9 注释与参考	204	11.7 注释与参考	305
8.10 进一步的阅读资料	204	11.8 进一步的阅读资料	305
习题	205	习题	306
第9章 近似算法	207	第12章 在线算法	307
9.1 顶点覆盖问题的近似算法	207	12.1 用贪心法解决在线欧几里得生成树问题	308
9.2 欧几里得旅行商问题的近似算法	208	12.2 在线k服务员问题及解决定义在平面上该问题的贪心算法	310
9.3 特殊瓶颈旅行商问题的近似算法	209	12.3 基于平衡策略的在线穿越障碍算法	315
9.4 特殊瓶颈加权k供应商问题的近似算法	213	12.4 用补偿策略求解在线二分匹配问题	322
9.5 装箱问题的近似算法	217	12.5 用适中策略解决在线m台机器调度问题	326
9.6 直线m中心问题的最优近似算法	218	12.6 基于排除策略的三个计算几何问题的在线算法	331
9.7 多序列比对问题的近似算法	220	12.7 基于随机策略的在线生成树算法	335
9.8 对换排序问题的2近似算法	225	12.8 注释与参考	338
9.9 多项式时间近似方案	230	12.9 进一步的阅读资料	339
9.10 最小路径代价生成树问题的2近似算法	239	习题	340
9.11 最小路径代价生成树问题的PTAS	241	参考文献	341
9.12 NPO完全性	245		
9.13 注释与参考	250		

# 第1章 绪 论

本书介绍算法的设计与分析，首先讨论下面这个非常重要的问题或许是有意义的：为什么要研究算法？通常认为是为了获得高速的计算，但这只要一台高速计算机就够了。这显然并不完全正确。我们通过一个实验来说明，实验结果清楚地表明：在低速计算机上执行好算法比在高速计算机上执行差算法完成得更好。考虑下面简单描述的两个排序算法：插入排序 (insertion sort) 和快速排序 (quick sort)。

插入排序是从左到右一个接一个地考查数据元素序列。在考查完每个元素之后，将元素插入已有序序列的一个适当位置。例如，假设将要排序的数据元素序列是

11, 7, 14, 1, 5, 9, 10

对上面的数据元素序列进行的插入排序如下：

有序序列	无序序列
11	7, 14, 1, 5, 9, 10
7, 11	14, 1, 5, 9, 10
7, 11, 14	1, 5, 9, 10
1, 7, 11, 14	5, 9, 10
1, 5, 7, 11, 14	9, 10
1, 5, 7, 9, 11, 14	10
1, 5, 7, 9, 10, 11, 14	

在上面的排序过程中，考虑当前有序序列是1, 5, 7, 11, 14的情况，下一个将要排序的元素是9。先将9与14比较，由于9比14小，接着将9与11比较。必须再一次继续比较，在将9与7比较之后，确定9比7大，因此，将9插入到7和11之间。

所举的第二个排序算法称为快速排序，将在第2章中作详细介绍。目前，只给出该算法一个较高层次的描述。设想对下列数据元素进行排序：

10, 5, 1, 17, 14, 8, 7, 26, 21, 3

快速排序选用第一个数据元素，也就是10，将整个数据元素分成三个子集合：一部分比10小，一部分比10大，以及等于10的部分。这样将数据集合表示成如下：

(5, 1, 8, 7, 3) (10) (17, 14, 26, 21)

现在必须对下面两个子序列排序：

(5, 1, 8, 7, 3)

和 (17, 14, 26, 21)

注意到它们是可以各自独立地进行排序的，划分的方法也可在两个子集上递归地实现。例如，考虑子序列

17, 14, 26, 21

使用17划分上面的数据集合，可以得到

(14) (17) (26, 21)

在将上面的序列26, 21排序成21, 26之后, 可以得到

14, 17, 21, 26

这是有序的序列。

相类似地,

(5, 1, 8, 7, 3)

可排序成

(1, 3, 5, 7, 8)

合并所有的有序序列, 可以得到

1, 3, 5, 7, 8, 10, 14, 17, 21, 26

这是最终有序的序列。

后面将证明快速排序比插入排序好得多。问题是: 有多好? 为了将快速排序与插入排序作比较, 我们在Intel 486上实现快速排序, 在IBM SP2上实现插入排序。IBM SP2是超级计算机, 在1997年曾击败过象棋大师, 而Intel 486只是普通的个人计算机。图1-1提供了实验结果, 对于每个点上的数字, 10个数据集合都是随机产生的, 并得到了两个算法的平均时间。可以看到当数据元素少于400时, 执行快速排序的Intel 486劣于执行插入排序的IBM SP2, 而当数据元素大于400时, Intel 486完成的要比IBM SP2好得多。

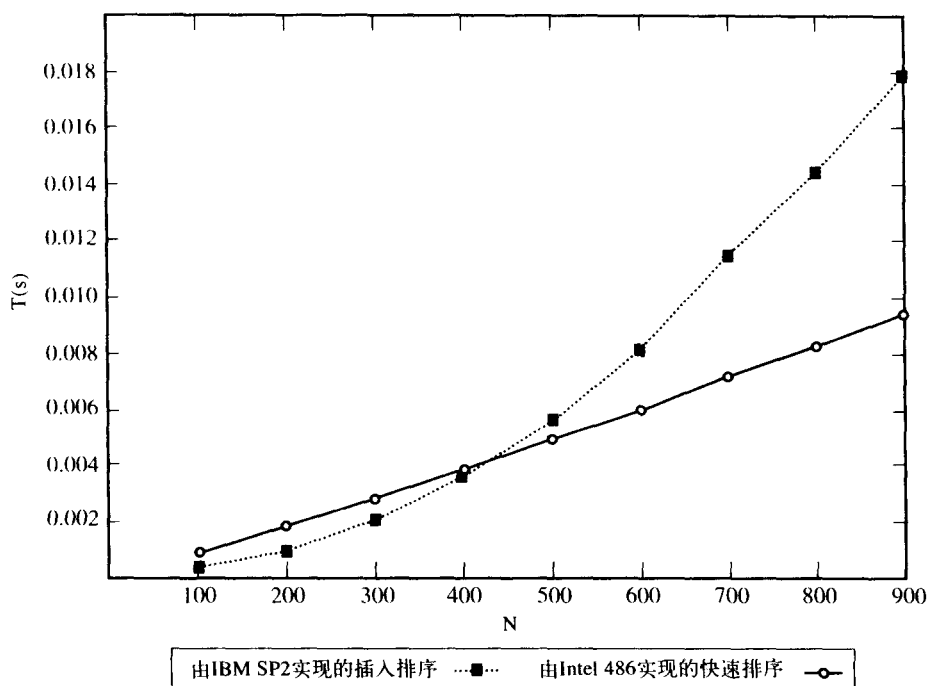


图1-1 插入排序与快速排序性能比较

上面的实验意味着什么呢? 显然它表明一个重要事实: 在高速计算机上执行低效算法比在低速计算机上执行高效算法差很多。换句话说, 如果你很有经验, 但没有好的算法知识, 那么你也也许不能够与一位对算法非常了解的笨拙同事相竞争。

如果承认研究算法是重要的, 就必须能够分析算法以确定它们的性能。在第2章中, 将对

算法分析相关的一些基本概念作简要介绍。在本章中,对算法分析在各种意义上都是介绍性的,在第2章末将介绍一些优秀的算法分析书籍。

在介绍了算法分析的概念后,将注意力转向问题的复杂度。我们要指出问题有易问题(easy problem)和难问题(difficult problem)之分。如果一个问题能够被有效算法解决,多半算法具有多项式时间(polynomial-time)复杂度,那么该问题称为易问题。反之,如果一个问题不能够被任何具有多项式时间的算法解决,那么它一定是难问题。通常,给定一个问题,如果已经知道存在多项式时间内解决该问题的算法,就可以确认它是一个易问题。然而,如果到目前为止还没有发现任何解决该问题的多项式时间算法,也不能推断在将来就找不到解决该问题的多项式算法。幸运的是,有一种用于度量问题复杂度的NP完全性(NP-completeness)理论。如果一个问题被证实是NP完全的(NP-complete),那么它被看作是个难问题,并且找到解决该问题的多项式算法的概率是非常小的。通常, NP完全性的概念都会在教科书的末尾才介绍。

许多问题看似不像是难问题,而事实上却是NP完全问题,这是非常有趣的。看下面一些例子。

首先,考虑0/1背包问题(0/1 knapsack problem)。这个问题非正式描述如下:假如我们要躲避入侵的军队必须离开心爱的家,随身携带一些有用的物品。但是,携带物品的总重量不能超过某个限制。在不超过重量限制的情况下,如何最大化所携带物品的价值呢?例如,假设有下列物品:

	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$
价值	10	5	1	9	3	4	11	17
重量	7	3	3	10	1	9	22	15

如果重量限制是14,那么最好的解是选择 $P_1, P_2, P_3$ 和 $P_5$ 。该问题是NP完全问题,随着物品的数目增大,很难找到一个最优解。

另一个看似容易的问题也是NP完全问题,这就是旅行商问题(traveling salesperson problem)。为了直观地解释该问题,考虑有许多城市,勤劳的推销商必须遍历每一座城市,每座城市不能遍历两次并且要求遍历路径最短。例如,一个旅行商问题实例的最优解如图1-2所示,其中两座城市间的距离是它们的欧几里得距离(Euclidean distance)。

最后,考虑分割问题(partition problem),即对于给定的整数集合,能否划分这些整数为两个子集合 $S_1$ 和 $S_2$ ,使得 $S_1$ 的和等于 $S_2$ 的和?例如,对于下面的集合

$$\{1, 7, 10, 9, 5, 8, 3, 13\}$$

可以划分上面的集合为

$$S_1 = \{1, 10, 9, 8\}$$

和  $S_2 = \{7, 5, 3, 13\}$ 。

可以验证 $S_1$ 的和等于 $S_2$ 的和。

我们经常需要解决这样的分割问题。例如,公钥加密方案就涉及这样的问题,这至今仍然是一个NP完全问题。

在这里还有另一个许多读者都会感兴趣的问题。我们都知道藏有许多无价珍宝的美术陈列馆,这些珍宝是不能被偷盗和损坏的,所以在陈列馆中安排有警卫。图1-3显示了一个美术陈列馆。

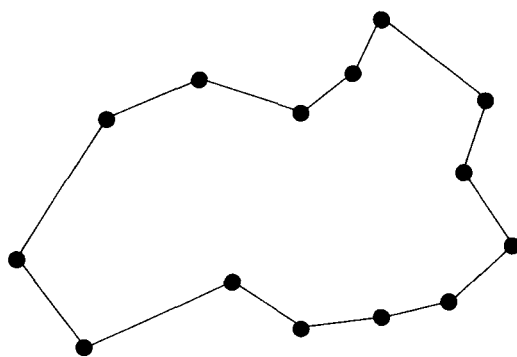


图1-2 一个旅行商问题实例的最优解

在图1-3中,如果在陈列馆中安排4名警卫,用小圆圈表示,陈列馆的每面墙至少由一名警卫监控。这意味着整个陈列馆被这4名警卫有效地监控。美术陈列馆问题 (art gallery problem) 是:给定一个多边形形状的陈列馆,确定最少的警卫数以及监控整个陈列馆的警卫所在的位置。令许多读者惊讶的是该问题也是一个NP完全问题。

即使意识到好算法是最本质的这个事实,人们仍然怀疑研究算法设计是否是重要的,因为好算法很容易获得。换句话说,如果好算法能够简单地通过直观或一般常识就可得到,那么就不值得费力去研究算法的设计。

下面在介绍最小生成树问题时,将会看到这个表面上看似组合爆炸的问题,实际上有非常简单的算法能够高效解决它。下面给出最小生成树问题的非形式化描述。

想象有许多城市,如图1-4所示。假设要连接所有城市成为一棵生成树 (spanning tree) (生成树是连接所有城市而没有环的图),使得生成树的总长度最小。例如,对于如图1-4所示的城市集合,其最小生成树 (minimum spanning tree) 如图1-5所示。

找出这样的最小生成树的直观算法是将所有可能的生成树都找出来,并确定每棵生成树的总长度。在穷举完所有可能的生成树之后,那么最小生成树就得到了。因为可能的生成树数量的确非常巨大,所以这样花费的代价也非常大。已经证明对于 $n$ 座城市,可能的生成树的总数目是 $n^{n-2}$ 。假如 $n = 10$ ,将要枚举 $10^8$ 棵树。当 $n = 100$ 时,这个数目将增加到 $100^{98}$ ( $10^{196}$ ),对于如此大的数字,是没有计算机可以处理的。在美国,电话公司可能必须处理 $n$ 为5 000的情况。因此,必须找到更好的算法。

事实上,解决这样的最小生成树问题有一个优秀的算法。我们通过举例来说明这个算法。如图1-6所示,算法首先找出 $d_{12}$ 是所有城市间的最短距离,所以连接城市1和2,如图1-7a所示。之后,将 $\{1, 2\}$ 作为一个集合,剩余的其他城市作为另一个集合。找出两个城市集合间的最短距离是 $d_{23}$ ,连接城市2和3。最后,连接城市3和4,整个过程如图1-7所示。

可以证明这个简单有效的算法总是能得到一个最优解,也就是该算法得到的生成树总是最小生成树。该算法的正确性证明并不容易,隐藏在此算法之后的策略称为贪心法 (greedy method)。通常,基于贪心法的算法是相当有效的。

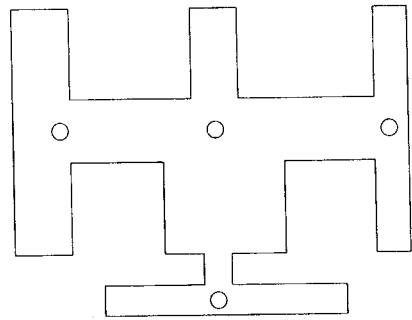


图1-3 美术陈列馆及安排的警卫

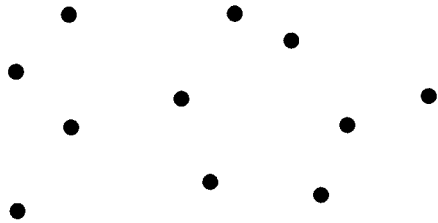


图1-4 说明最小生成树问题的城市集合

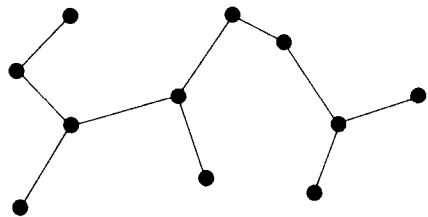


图1-5 对于图1-4所示城市集合的一棵最小生成树

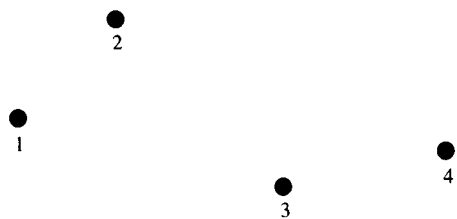


图1-6 说明一个有效的最小生成树算法的例子



遗憾的是，许多类似于最小生成树的问题却不能使用贪心法解决。例如，旅行商问题就是这样的。

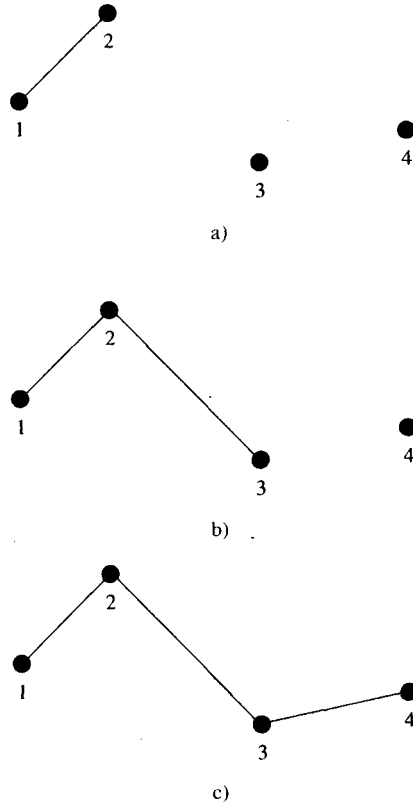


图1-7 最小生成树算法举例

不易使用穷举搜索的另一个例子是1中心问题 (1-center problem)。在1中心问题中，给定一个点集合，必须找到一个覆盖所有点的圆，使圆的半径最小。例如，图1-8显示了一个1中心问题的最优解。对于该问题如何开始解答呢？本书的后续章节将说明1中心问题可以使用剪枝搜索策略 (prune-and-search strategy) 来解决。

算法设计的研究几乎就是对策略的研究。由于众多研究者的努力，发现了许多优秀的策略，这些策略可以用于设计算法。我们不能断言每个优秀的算法一定是基于某个一般性的策略，但是，可以明确地说完备的策略知识对于算法的研究绝对是非常重要的。

现在，推荐一些关于算法的书目，它们适合于作为进一步阅读的参考。

Aho, A. V., Hopcroft, J. E. and Ullman, J. D. (1974):

*The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass.

Basse, S. and Van Gelder, A. (2000): *Computer Algorithms: Introduction to Design and*

*Analysis*, Addison-Wesley, Reading, Mass.

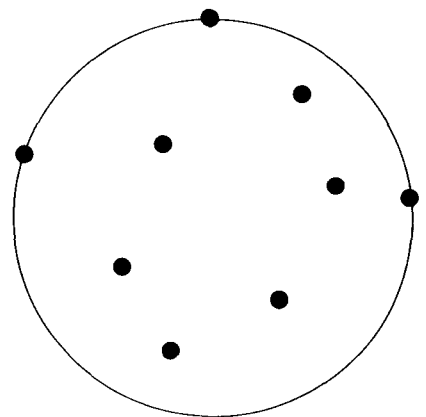


图1-8 1中心问题的解

- Brassard, G. and Bratley, P. (1988): *Algorithmics: Theory and Practice*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Coffman, E. G. and Lueker, G. S. (1991): *Probabilistic Analysis of Packaging & Partitioning Algorithms*, John Wiley & Sons, New York.
- Cormen, T. H. (2001): *Introduction to Algorithms*, McGraw-Hill, New York.
- Cormen, T. H., Leiserson, C. E. and Rivest, R. L. (1990): *Introduction to Algorithms*, McGraw-Hill, New York.
- Dolan A. and Aldous J. (1993): *Networks and Algorithms: An Introductory Approach*, John Wiley & Sons, New York.
- Evans, J. R. and Minieka, E. (1992): *Optimization Algorithms for Networks and Graphs*, 2nd ed., Marcel Dekker, New York.
- Garey, M. R. and Johnson, D. S. (1979): *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, California.
- Gonnet, G. H. (1983): *Handbook of Algorithms and Data Structures*, Addison-Wesley, Reading, Mass.
- Goodman, S. and Hedetniemi, S. (1980): *Introduction to the Design and Analysis of Algorithms*, McGraw-Hill, New York.
- Gould, R. (1988): *Graph Theory*, Benjamin Cummings, Redwood City, California.
- Greene, D. H. and Knuth, D. E. (1981): *Mathematics for the Analysis of Algorithms*, Birkhäuser, Boston, Mass.
- Hofri, M. (1987): *Probabilistic Analysis of Algorithms*, Springer-Verlag, New York.
- Horowitz, E. and Sahni, S. (1976): *Fundamentals of Data Structures*, Computer Science Press, Rockville, Maryland.
- Horowitz, E., Sahni, S. and Rajasekaran, S. (1998): *Computer Algorithms*, W. H. Freeman, New York.
- King, T. (1992): *Dynamic Data Structures: Theory and Applications*, Academic Press, London.
- Knuth, D. E. (1969): *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*, Addison-Wesley, Reading, Mass.
- Knuth, D. E. (1973): *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Addison-Wesley, Reading, Mass.
- Kozen, D. C. (1992): *The Design and Analysis of Algorithms*, Springer-Verlag, New York.
- Kronsjö, L. I. (1987): *Algorithms: Their Complexity and Efficiency*, John Wiley & Sons, New York.
- Kucera, L. (1991): *Combinatorial Algorithms*, IOP Publishing, Philadelphia.
- Lewis, H. R. and Denenberg, L. (1991): *Data Structures and Their Algorithms*, Harper Collins, New York.
- Manber, U. (1989): *Introduction to Algorithms: A Creative Approach*, Addison-Wesley, Reading, Mass.
- Mehlhorn, K. (1987): *Data Structures and Algorithms: Sorting and Searching*, Springer-Verlag, New York.
- Moret, B. M. E. and Shapiro, H. D. (1991): *Algorithms from P to NP*, Benjamin Cummings, Redwood City, California.